REMARKS

Status of the Claims

- Claims 1-6 and 8-23 are pending in the Application.
- Claims 1-6 and 8-23 are rejected by Examiner.

Claim Rejections Pursuant to 35 U.S.C. §102

Examiner has rejected Claims 1-6 and 8-23 under 35 U.S.C. §102(e) as being anticipated by U.S. Pat. No. 7,120,645 to Manikutty et al. (Manikutty). Applicant respectfully traverse the §102(e) rejection.

Manikutty at col. 5, lines 13-25 teaches a technique for executing database commands that involve operations on XML constructs includes receiving the database command. It is then determined whether an XML component operation in the database command can be transformed to a relational database operation, which operates on a particular set of one or more relational database constructs, and which does not involve the XML component operation. If it is determined that the XML operation can be transformed, then the XML component operation is rewritten to a particular relational database operation that does not involve the XML component operation. The particular relational database operation on the particular set of one or more relational database constructs is evaluated. (see col. 5, lines 13-25).

Figure 3 of Manikutty is a flow diagram that illustrates a method of rewriting a database query that has an XML operation. (see col. 4, lines 52-54). Figure 4 of Manikutty is a flow diagram of step 320 of Figure 3. With regard to Step 450 of Figure 4, Manikutty teaches at col. 20 line 61- col. 21, line 17:

"Tree of Canonical XML Generation Functions

In step 450, canonical XML generation functions are expanded to a normalized tree of canonical functions. In other embodiments, the primitive XML generation operations are expanded to a normalized tree of primitive operations. In a normalized tree, each XEL function has a single non-XML type operand, such as a single scalar operand, or a single abstract data type (ADT) operand, or a single collection type column

(typically at a leaf node of the normalized tree), or a set of operands that are of XML type (typically at a parent node of the normalized tree). In some embodiments, a set of operands of XML type might be the result of a tree of canonical XML generation functions. It is ensured that each child in the normalized tree is explicitly tagged. If the child is an ADT that is not tagged, the XEL function is converted to the form XEL (null, "<value expression>" as "<type name>"). Scalar operands are always tagged. For example, FIG. 5 is a block diagram that illustrates a normalized tree 510 of XML generation operations in the sub-query for view dept_xv. In tree 510, each XEL function has a single non-XML type operand, or a set of XML type operands." (col. 20, line 61 through col. 21, line 17).

Thus, Applicant notes that Manikutty relies on a tree structure to map in canonical XML generation functions. Applicant also notes that Manikutty is absent any teaching of the use of a graph structure.

Claim 1 recites, in relevant part:

"1. A method for semantic representation of one or more XML language inquiries across relational and non-relational data sources comprising:

receiving…;

defining a plurality of nodes *of a graph structure which represents the at least one inquiry, the graph structure having at least one node object for every operation within the at least one received inquiry*;

translating…; and

generating a semantic representation *having the graph structure*…"
(Part of Claim 1)

Applicant notes that Claim 1 relies on a graph structure for the semantic representation. As a reference, the as-filed specification states, in paragraph 0046, that:

> "The XML intermediate language is a representation of an XML query or view. As such, it may be termed an query intermediate language (QIL) because it is an explicit representation of the meaning of an XML query. The QIL may be viewed as a semantic representation across XML query query and view language compilers. QIL is similar to an ordinary abstract syntax tree (AST) but different in that QIL captures not the syntax but the semantics, or meaning, of a query. Another difference is that *QIL is a graph structure and not a tree structure* like AST." (present application, para. 0046)

Applicant respectfully submits that the "graph structure", as used in Claim 1 is specifically not a tree structure because it is so defined in the as-filed specification.

> The as-filed specification of the present application at paragraph 0049 states:
> "The query intermediate language utilizes its own operators with *nodes of a graph structure* which represents the original query". (present application, para. 0049)

Whereas Manikutty explicitly relies on a tree structure to express canonical XML generation functions (see cols 20-21), Claim 1 explicitly recites the use of a graph structure which is defined in the as-filed specification as explicitly not a tree structure.

Accordingly, Manikutty fails to teach every element of Claim 1. For instance, because Manikutty fails to teach a graph structure, Manikutty fails to teach the Claim 1 elements of "defining a plurality of nodes of a graph structure which represents the at least one inquiry, the graph structure having at least one node object for every operation within the at least one received inquiry", and "generating a semantic representation having the graph structure".

Applicant notes that independent Claims 11, 17, and 21 also recite elements having a graph structure which is not taught in Manikutty.

Since Manikutty fails to teach that a semantic representation includes a graphical structure, and since a graph structure is explicitly taught in the as-filed specification as not being a tree structure upon which Manikutty relies, then Manikutty cannot anticipate independent Claims 1, 11, 17, and 21 and their respective dependent claims.

Applicant therefore respectfully requests withdrawal of the 35 USC §102(e) rejection and submits that Claims 1-6, and 8-23 patentably define over the cited art because all elements of the independent Claims 1, 11, 17 and 21 are not found in the cited art.

Conclusion

Applicant respectfully requests reconsideration of all pending claims in light of the amendment and discussion above. Applicant respectfully submits that all pending claims patentably define over the cited art and respectfully requests a Notice of Allowance for all pending claims.

Respectfully Submitted,

Date: May 9, 2007

/Jerome G. Schaefer/

Jerome G. Schaefer
Registration No. 50,800

Woodcock Washburn LLP
Cira Centre
2929 Arch Street, 12th Floor
Philadelphia, PA 19104-2891
Telephone: (215) 568-3100
Facsimile: (215) 568-3439